Laravel is a rich web application framework like Symfony. It features ‘expressive, elegant syntax’ intended for the development of web applications following the model–view–controller (MVC) architectural pattern. Laravel believes that development must be an enjoyable and creative experience for the programmers and web designers for it to be truly fulfilling. Laravel is based on the Symfony framework. A CMS called Drupal also inherits many components from Symfony. However, Laravel is more advanced than Symfony and programmers and web designers find it to be much easier for their platforms.

In this article, we will be using composer to create a Laravel application on CyberPanel.

For installing Symfony on CyberPanel, please refer to this article [Using Composer create Laravel application on CyberPanel!](https://cyberpersons.com/2017/12/14/using-composer-create-symfony-application-on-cyberpanel/)

## Step 1: Install CyberPanel!

First, you need to get CyberPanel installed on your server. CyberPanel is a web hosting control panel based on OpenLiteSpeed web server.

You will need a fresh install of CentOS 7.x. In this article, we will be using a fresh and updated install of Centos 7.3.

To install CyberPanel, enter these commands, for more details visit [installation](http://docs.cyberpanel.net/doku.php?id=installation).

wget http://cyberpanel.net/install.tar.gz

tar zxf install.tar.gz

cd install

chmod +x install.py

python install.py [IP Address]

Replace [IP Address] with your vps/dedicated server IP Address.

## Step 2: Install Composer!

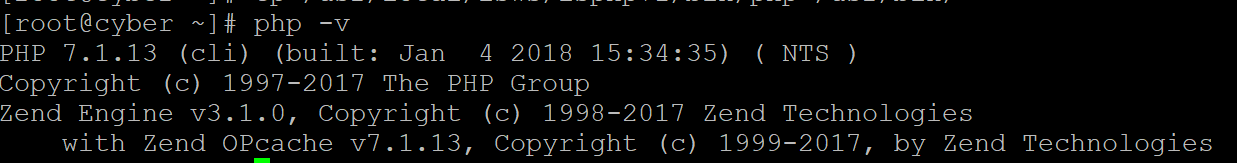
To install Composer you need php-cli. CyberPanel comes with PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1. We will use php-cli from PHP 7.1 because Symfony 4 requires php 7.1 and above. You can copy PHP 7.1 binary, so that it is available globally.

cp /usr/local/lsws/lsphp71/bin/php /usr/bin/

With this, you can use php-cli by just trying ‘php’. You can verify this using ‘php -v’.

php -v

You should get :



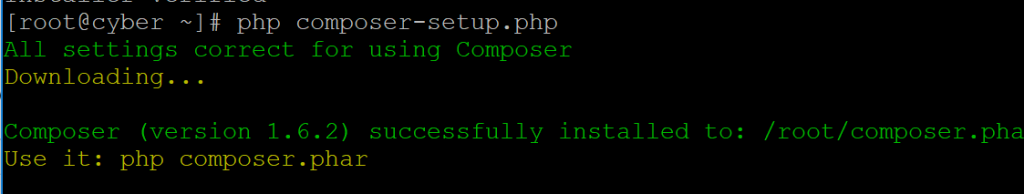
That means you are ready to install composer. To install composer use these commands :

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

php -r "if (hash\_file('SHA384', 'composer-setup.php') === '544e09ee996cdf60ece3804abc52599c22b1f40f4323403c44d44fdfdd586475ca9813a858088ffbc1f233e9b180f061') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP\_EOL;"

php composer-setup.php

php -r "unlink('composer-setup.php');"



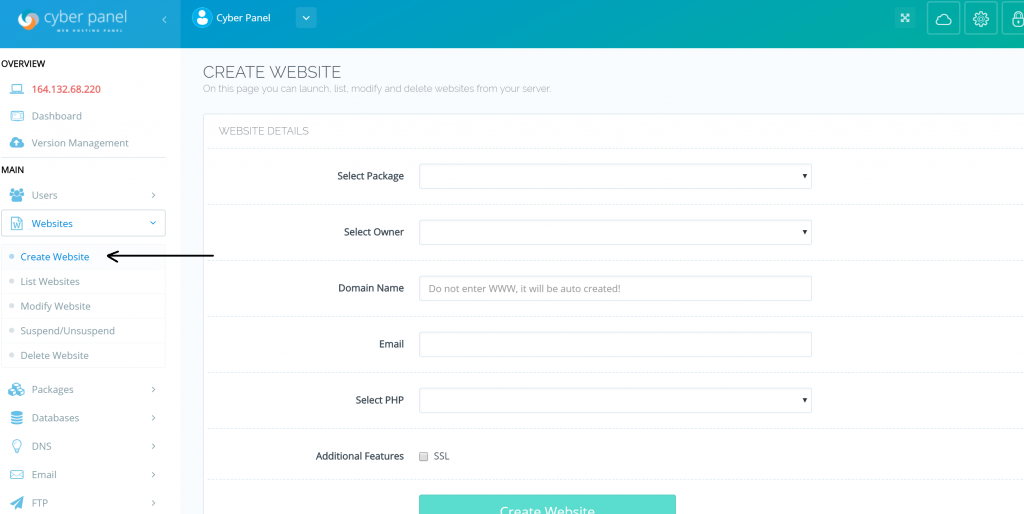
You can use composer project-wise, but for convenience, we will copy the composer binary to a global location using:

cp composer.phar /usr/bin/composer

After this command, composer is now available globally and you can trigger it just using the ‘composer’ command. With all of this work done, we are now ready to create our first site and launch a Laravel application!

## Step 3: Create Website!

To launch your Larevel application, you need to create a website in your CyberPanel. To login to CyberPanel, go to [Server IP]:8090 (the port should be open by firewall) and login as admin with the default password 1234567. It is HIGHLY recommended to change your password after doing so via Edit Profile.



From left menu click ‘Create Website’, enter your domain name and the other details and select ‘PHP 7.1’ because Laravel framework requires PHP 7.0 and above. As PHP 7.1 is available as of the time of this writing, we will be using that.

Once the website is successfully created you can use composer to create a Laravel web application.

For setting up the laravel framework and creating a project, SSH access to the server is required. As also mentioned in the previous article, the directory of every website in CyberPanel looks like /home/example.com/public\_html/, where “example.com” is a substitute for the actual domain name you will be using.

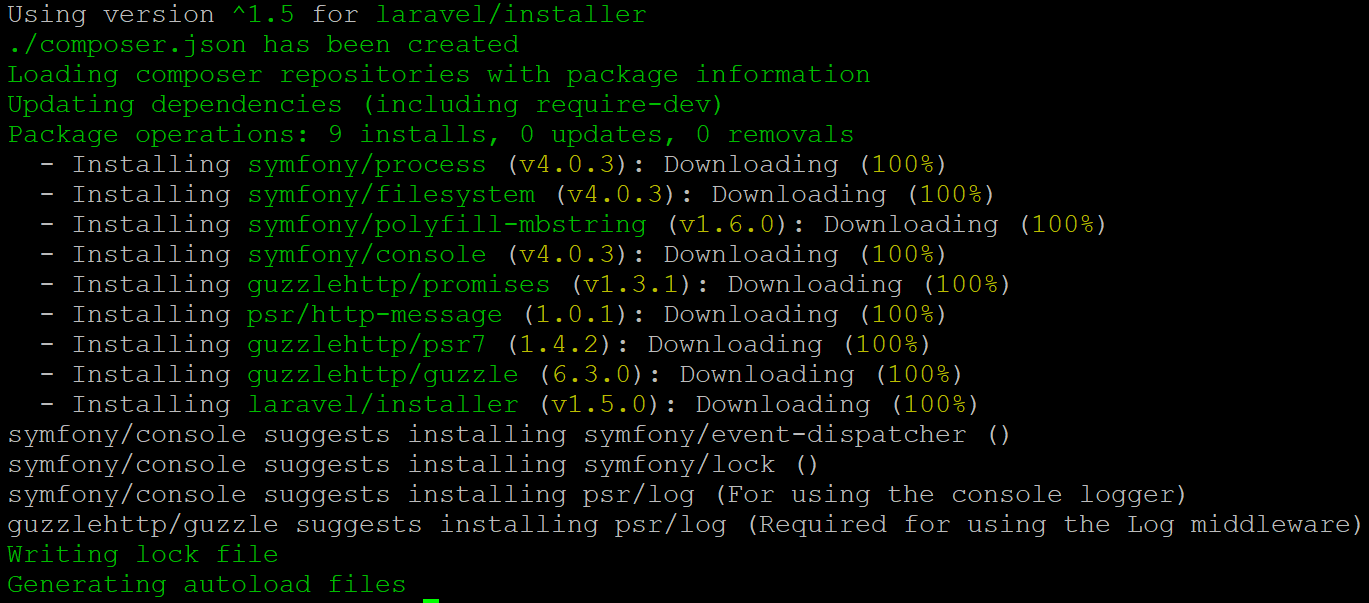
## Step 4: Downloading Laravel!

Following this directory structure, we will now create our application.

cd /home/example.com/public\_html

composer global require "laravel/installer"

Now, composer will download the basic packages and dependencies required and Laravel itself. You should see something like this on the SSH console :



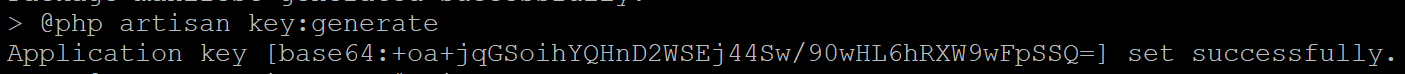
## Step 5: Setting up our Laravel Project!

After this, Laravel installer will be downloaded and we will be able to make our projects with Laravel using this command :

composer create-project --prefer-dist laravel/laravel cyber



Composer should now install Laravel and will set up our project using the appropriate directory structure inside our web domain. Please note that cyber is the project name we are using for this article. You can replace ‘cyber’ with the project name that you would like to give to your application. After this our Laravel application should be installed.

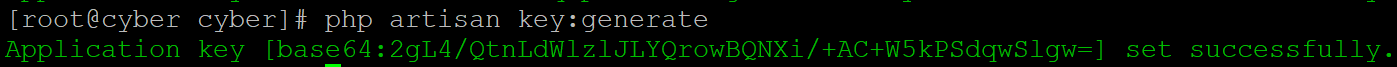


At the end of the installation and setup being done by Composer, we will see an Application key. This key will be used while creating Laravel applications. You may wish to note it or copy it down somewhere. You can also generate a new App Key by browsing to our project directory and using the command :

cd public

php artisan key:generate

As expected, a new App key will be generated for us.



## Step 6 : Making the directories writable

Even though our application has been set up, it will not function on the web server. It gives errors because the directories for Laravel are not yet writable and therefore our application won’t function. To make the directories writable, we will enter these commands :

chmod -R 777 storage

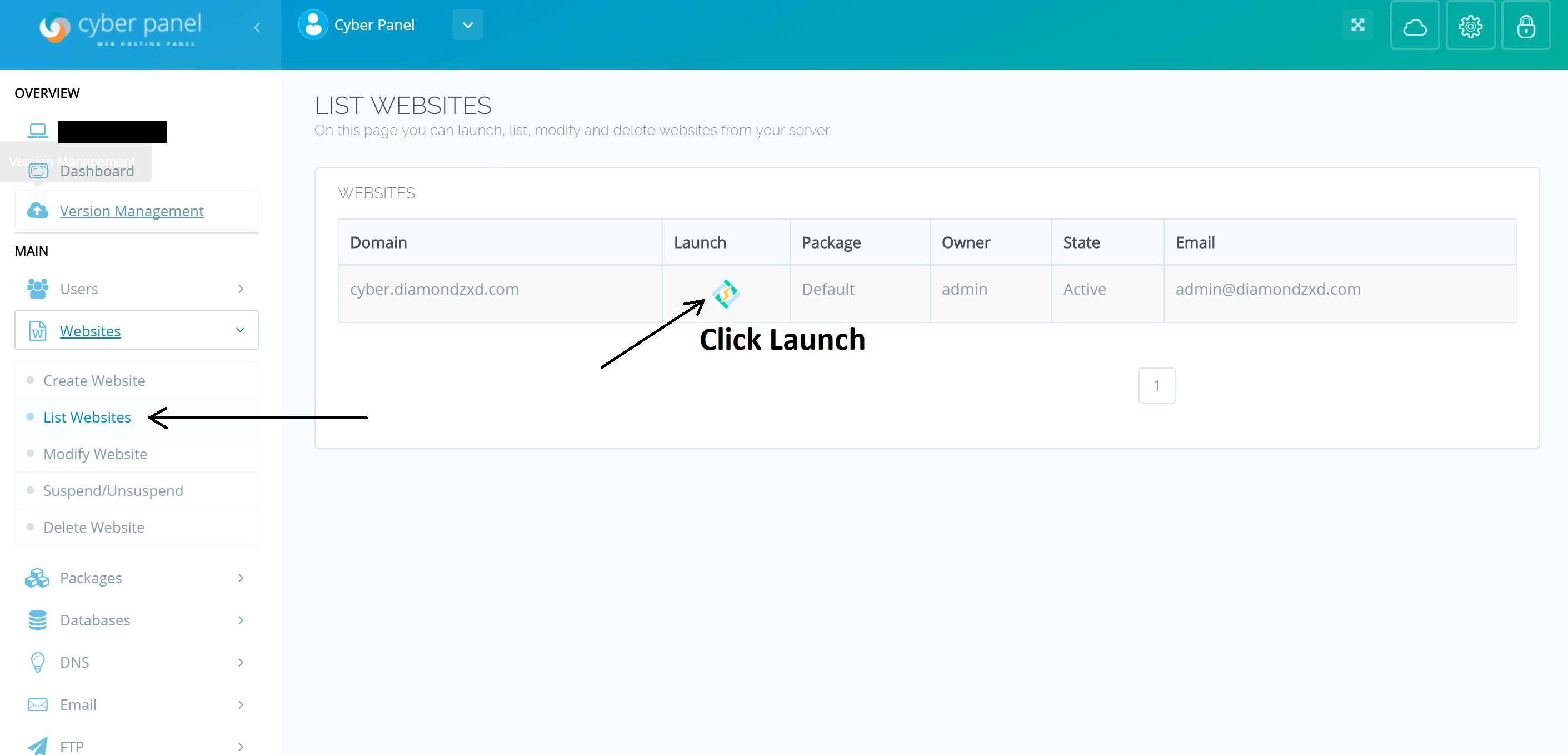
chmod -R 777 bootstrap/cache

Congratulations! Your Laravel Application should be functioning now inside http://example.com/[your project name]/public

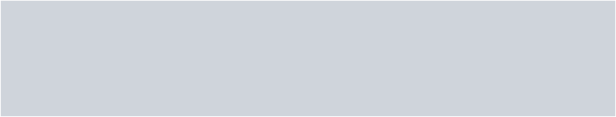


## Step 7 : Changing Rewrite Rules

To make URL’s more friendly we will now change the Rewrite (.htaccess) rules. To open your domain’s .htaccess file, go to the Left menu of CyberPanel and select ‘List Websites’. Find your domain and click on Launch.



Scroll down through the Panel options and click on “Add Rewrite Rules (.htaccess)” option.



Since CyberPanel is based on OpenLiteSpeed webserver, your websites will read Apache’s .htaccess perfectly! This makes OpenLiteSpeed server and CyberPanel unique and faster compared to others. Now we will add the following Rewrite rules for our domain to have friendly URL’s and to make Laravel work on the root of your domain :

RewriteEngine On

RewriteRule . /cyber/public/index.php [L]

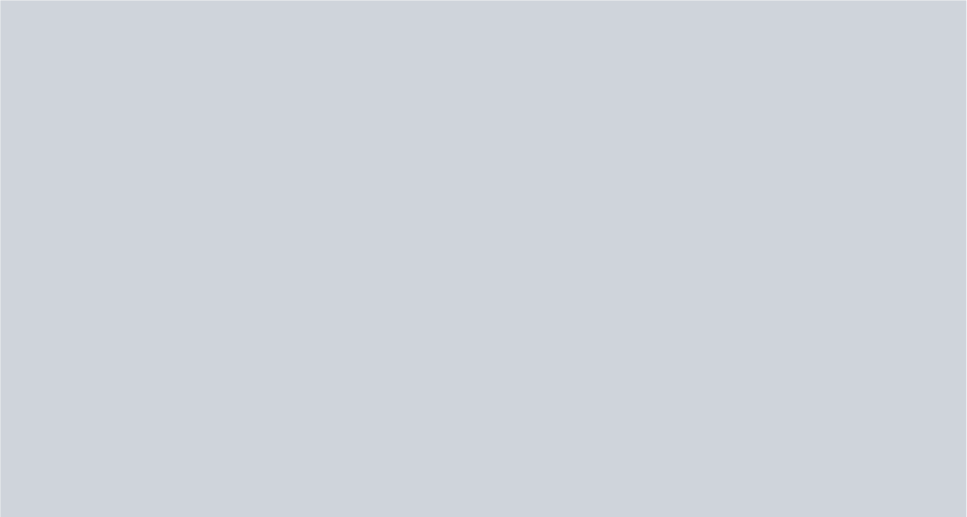
RewriteCond %{REQUEST\_FILENAME} !-d

RewriteCond %{REQUEST\_FILENAME} !-f

Please remember that “cyber” is the project name we used for this article. Please substitute the project name you would like to use with your own application!

After adding the rules, click on ‘Save Rewrite Rules’.

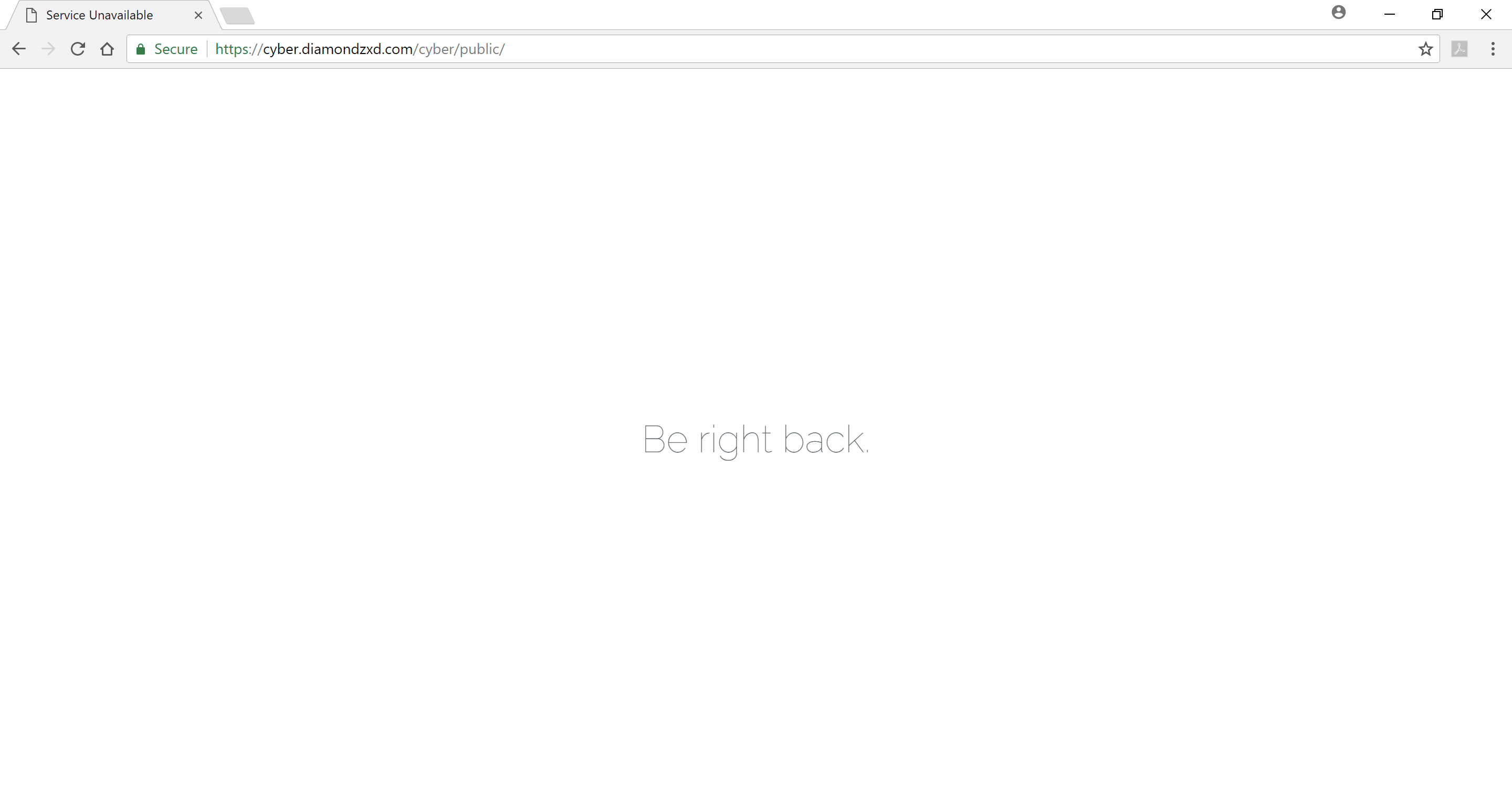
Congratulations, now your Laravel application should be fully functional and will be working on the root of your domain as well!



You can fiddle with some basic Laravel commands in your project directory to see its working.

php artisan down

This command will put your application in maintenance mode



to bring it back online you can use this command and your site will be back up and functional:

php artisan up

We hope you liked this article, don’t forget to comment and share! For more posts like these, stay connected to this website for more Tips and Tricks. If you would want to know more anything about a technology or just something from this article, don’t hesitate to comment down and we sure will reply to your concern.

https://cyberpanel.net/blog/2018/01/22/how-to-setup-laravel-application-on-cyberpanel/